

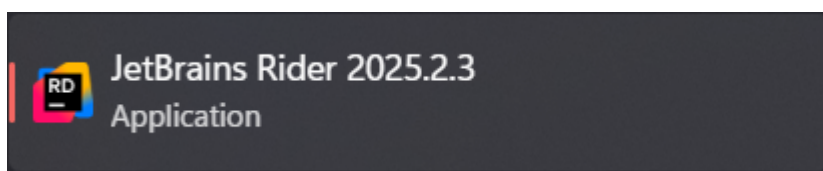
Mise en place de l'environnement C#

Avant d'écrire du code, on installe les bons outils et on comprend comment ils s'organisent. Cette étape ne prend que quelques minutes et servira pour toute la suite du cours.

Les outils nécessaires

Pour développer en C# sous Windows, on a besoin de deux choses :

Outil	Rôle
.NET SDK	La "machinerie" qui compile et exécute le code C#
JetBrains Rider	L'éditeur intelligent (IDE) dans lequel on code



Il existe d'autres outils pour programmer en C# (Visual Studio, VS Code...). Je recommande Rider pour ce cours.

Installer Rider

Licence gratuite pour les étudiants

JetBrains propose une licence éducation **entièrement gratuite** pour les élèves et enseignants.

👉 Crée ton compte sur jetbrains.com/student avec ton adresse mail scolaire.

Téléchargement

👉 jetbrains.com/rider → Windows → Download

Lance l'installateur et accepte les options par défaut.

Le .NET SDK

Le SDK (*Software Development Kit*) est l'ensemble des outils qui permettent de **compiler** et **exécuter** du code C#.

💡 **Bonne nouvelle** : tu n'as pas besoin de l'installer séparément à l'avance.

Quand tu crées un nouveau projet dans Rider, il détecte automatiquement si le SDK requis est présent – et te

propose de le **télécharger directement** depuis l'interface si ce n'est pas le cas.

Si tu préfères l'installer manuellement :

👉 dotnet.microsoft.com/download → choisis la version **LTS** (Long Term Support)

Pour vérifier qu'il est bien installé, ouvre un terminal et tape :

```
1 | dotnet --version
```

Tu dois voir quelque chose comme **8.0.x**.

Solution et Projet : comment ça s'organise ?

Avant de créer quoi que ce soit, il faut comprendre deux concepts clés dans Rider (et dans tout l'écosystème .NET) :

La Solution

Une **Solution** est le **conteneur global** de ton travail. Elle regroupe un ou plusieurs projets.

Concrètement, c'est un fichier **.sln** qui sert de point d'entrée à Rider.

Le Projet

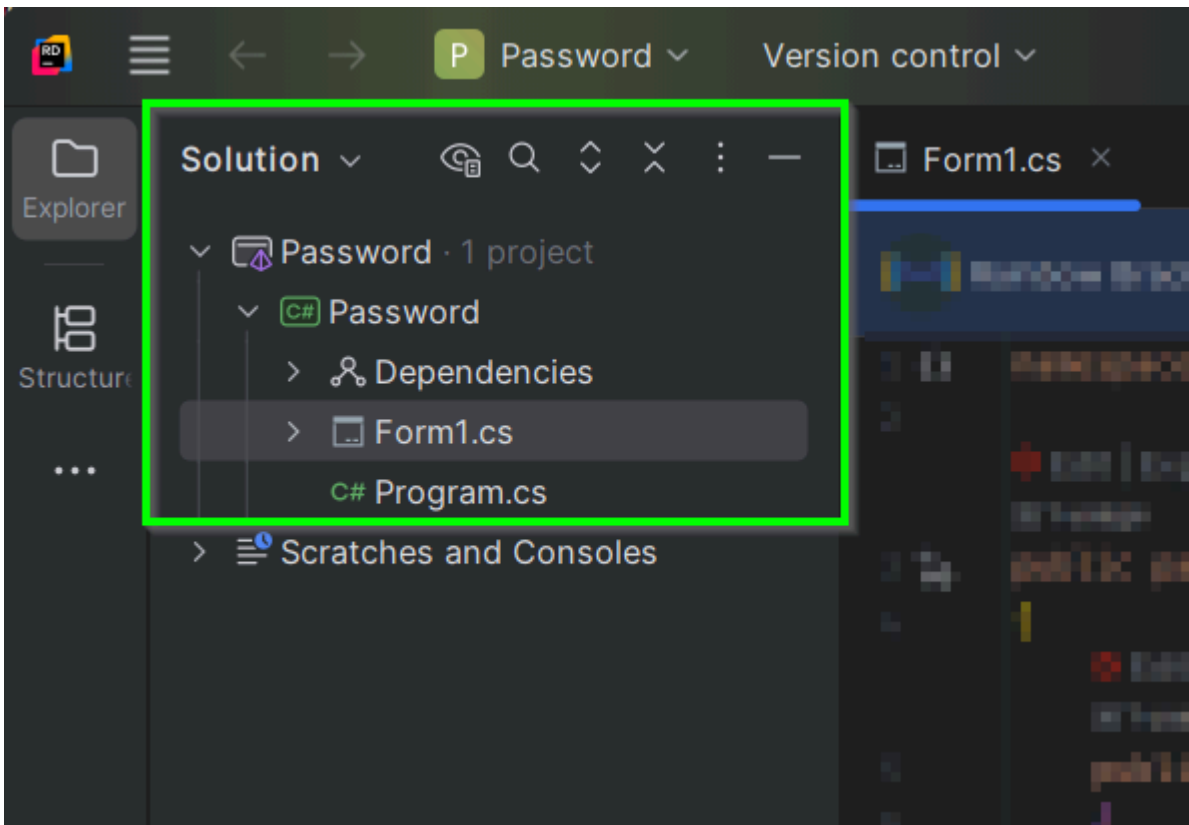
Un **Projet** est une **application autonome** à l'intérieur de la solution — avec son propre code, ses propres fichiers, et un fichier **.csproj** qui le décrit.

```
MaSolution/                ← Solution (.sln)
├── MonApplication/        ← Projet (.csproj)
│   └── Program.cs
└── AutreApplication/     ← 2e Projet (si besoin)
    └── Program.cs
```

En cours, on aura souvent **une solution = un projet**. Mais dès qu'on aborde la POO, on verra qu'il peut être utile de séparer les choses.

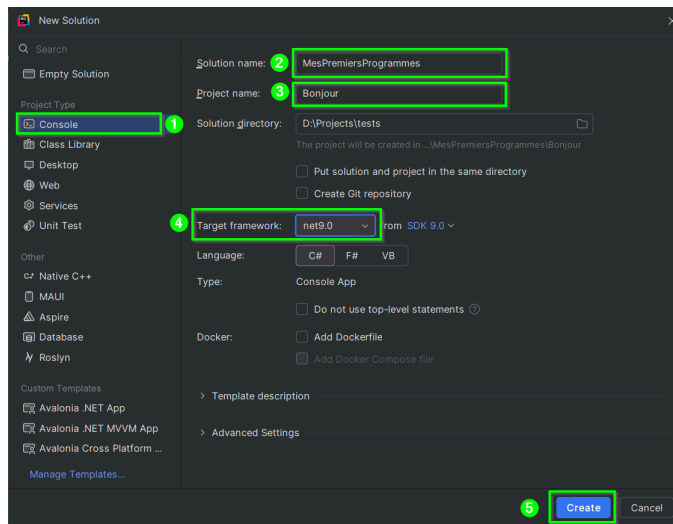
Exemple

Tu as ici une solution "Password" qui contient un projet "Password", qui contient à son tour des fichiers c-sharp:



Créer un premier projet

Étape par étape



- Lance Rider → clique sur **New Solution**
- Dans la liste de gauche, choisis **Console Application** [1]
- Remplis les champs :
 - **Solution name** : **MesPremiersProgrammes** (pas d'espaces, pas d'accents) [2]
 - **Project name** : **Bonjour** [3]
 - **Location** : ton dossier de travail
 - **Target Framework** : choisis la version **LTS** disponible (ex: **.NET 9**) [4]

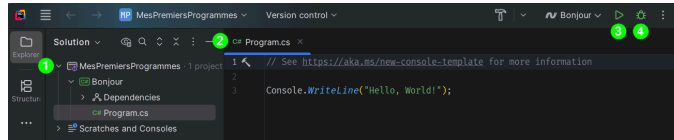
💡 Si le SDK n'est pas installé, Rider affiche un avertissement à cette étape avec un bouton **Download .NET SDK**. Clique dessus – il s'installe automatiquement.

- Clique sur **Create** [5]

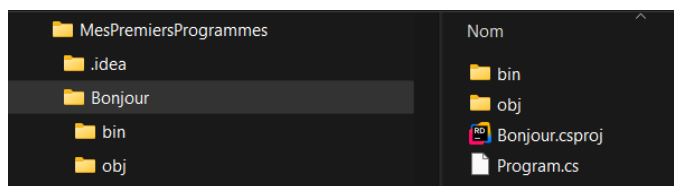
Comprendre l'arborescence générée

Une fois le projet créé:

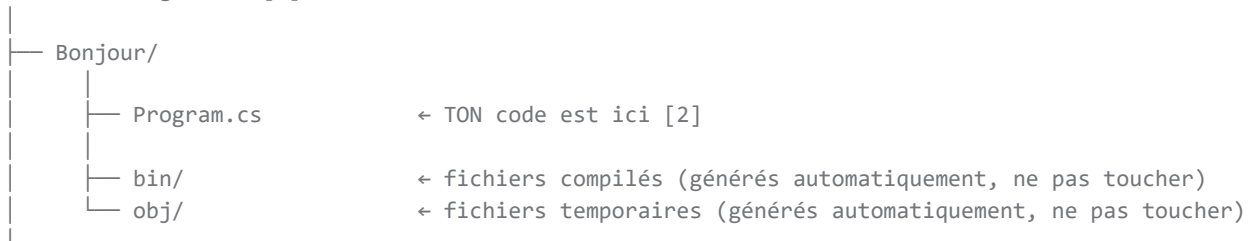
Rider affiche ceci:



Sur ton disque, tu auras la structure suivante:



MesPremiersProgrammes [1]



Le seul fichier que tu modifies au quotidien : **Program.cs**.

Exécuter un programme

Rider propose deux façons de lancer ton programme :

Action	Raccourci	Quand l'utiliser
Exécuter [3]	Shift + F10	Pour voir le résultat
Débugger [4]	Shift + F9	Pour inspecter pas à pas

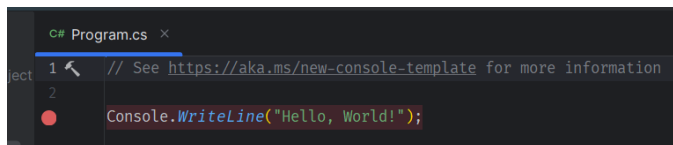
La sortie apparaît dans le panneau **Run** en bas de l'écran.

Le débogueur : ton meilleur allié

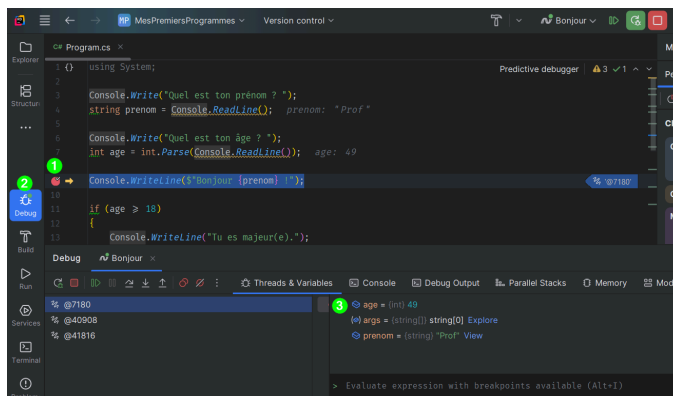
Le débogueur permet de **mettre le programme en pause** à une ligne précise pour inspecter les valeurs des variables.

Poser un breakpoint

Clique dans la marge à gauche d'une ligne → un point rouge ● apparaît.



Lance ensuite en mode **Debug** (**Shift + F9**).



Le programme s'arrête à cette ligne. Tu peux voir dans le panneau **Debug** [2] la valeur de chaque variable à cet instant.

Avancer pas à pas

Raccourci	Action
F8	Exécuter la ligne et passer à la suivante
F7	Entrer dans une méthode
F9	Continuer jusqu'au prochain breakpoint

Raccourcis essentiels

À connaître dès le premier jour :

Raccourci	Action
Shift + F10	Exécuter

Raccourci	Action
Shift + F9	Déboguer
F9	Poser / enlever un breakpoint
F8	Pas à pas
Ctrl + /	Commenter / décommenter une ligne
Alt + Entrée	Correction / suggestion automatique
Ctrl + Z	Annuler
Ctrl + Shift + Z	Rétablir

💡 Alt + Entrée est probablement le raccourci le plus puissant de Rider. Quand tu ne sais pas quoi faire face à une erreur, essaie-le.



Exercice – Vérifier que tout fonctionne

Objectif : confirmer que Rider et .NET sont correctement installés.

Consigne : dans `Program.cs`, écris ce code et exécute-le :

```
1 | Console.WriteLine("Rider est prêt !");
2 | Console.WriteLine($"Version .NET : {Environment.Version}");
```

Résultat attendu :

```
Rider est prêt !
Version .NET : 8.0.x
```

Si tu vois ces deux lignes dans le panneau Run : **ton environnement est opérationnel** ✓

À retenir

- Le **.NET SDK** compile et exécute le code – Rider peut le télécharger pour toi
- **Rider** est l'IDE – licence gratuite avec ton adresse scolaire
- Une **Solution** contient un ou plusieurs **Projets**
- `Program.cs` est le fichier où tu codes
- `bin/` et `obj/` sont générés automatiquement – ne pas y toucher
- `Shift + F10` pour exécuter, `Shift + F9` pour déboguer

```
1 ---
2
3 ## Alternative – Visual Studio Code
4
5 Si Rider n'est pas disponible sur ta machine, tu peux coder en C# avec Visual Studio Code,
6
7 ### Installation
8
9 1. 🖱️ [code.visualstudio.com](https://code.visualstudio.com) → télécharge et installe VS Code
10 2. Dans VS Code, ouvre le panneau Extensions (Ctrl + Shift + X)
11 3. Recherche et installe C# Dev Kit (éditeur : Microsoft)
12
13 Le .NET SDK reste nécessaire – installe-le depuis [dotnet.microsoft.com/download](https://dotnet.microsoft.com/download)
14
15 ### Ce que tu y trouveras
16
17 - Coloration syntaxique et autocomplétion
18 - Exécution et débogage intégrés
19 - Gestion de la Solution et du Projet via un panneau dédié
20
21 ### Ce que tu n'y trouveras pas
22
23 VS Code reste moins confortable que Rider pour le C# : le refactoring automatique, le débogueur
24
25 > 💡 Si tu connais déjà VS Code pour PHP ou JavaScript, tu t'y sentiras à l'aise – mais investis
```